

CLAIMS

1. (currently amended) A method ~~of packaging~~ creating a dynamically linked computer program function ~~package~~, the method comprising:

establishing an attribute, each attribute exhibiting a plurality of at least one of variations, characteristics and parameters associated with said ~~dynamically-linked~~ a specific computer program function;

obtaining a source file associated with said ~~dynamically-linked~~ specific computer program function;

compiling said source file iteratively to create a plurality of corresponding object files based on said at least one of variations, characteristics, and parameters for each said attribute, wherein each iteration of the compiling said source file results in the plurality of object files having the specific computer program function but with each object file having distinct attributes from one another; and

linking ~~a the~~ the plurality of ~~intermediately-resulting~~ object files to create a single executable file, such that the single executable file comprises different versions of the specific computer program function; and

wherein said single executable file is configured to facilitate choice of a selected version of said specific computer function based on a particular said at least one of variations, characteristics, and parameters for each said attribute.

2. (currently amended) The method of Claim 1 further including configuring an application to be responsive to said selected version of said specific computer program function based on said particular at least one of variations, characteristics, and parameters for each said attribute.

3. (original) The method of Claim 2 wherein said configuring includes compiling based on said particular at least one of variations, characteristics, and parameters for each said attribute.

4. (original) The method of Claim 1 wherein said attribute includes: version, addressability, character code base, character set that a specific operating system supports, system linkage conventions; machine architecture, or floating point hardware.

5. (original) The method of Claim 1 wherein said attribute is user specified.

6. (original) The method of Claim 1 wherein said attribute is implicitly defined.

7. (original) The method of Claim 1 wherein said at least one of variations, characteristics, and parameters for each said attribute includes, 64-bit versus 32-bit addressing; ASCII versus EBCDIC, internal representation for character data, or HEX versus IEEE representation to use for floating point data.

8. (original) The method of Claim 1 wherein said at least one of variations, characteristics, and parameters for each said attribute is user specified.

9-11. (cancelled)